

**METHOD ENABLING REAL-TIME TESTING OF ON-DEMAND  
INFRASTRUCTURE TO PREDICT SERVICE LEVEL AGREEMENT  
COMPLIANCE**

5

**Technical Field**

[0001] The present invention relates generally to electronic business on demand  
10 (EBOD) and, more specifically, to a system and method for providing a real-time  
infrastructure scenario testing within an EBOD environment.

**Background of the Invention**

15 [0002] For decades, International Business Machines Corp. (IBM) of Armonk, New  
York has been at the forefront of new paradigms in business computing. Currently  
IBM is developing a new computing utility service initiative, or "e-business on  
demand" (EBOD). Briefly, EBOD is a form of information technology (IT) based  
upon "power by the hour" in which a client pays only for the level of computing  
20 services actually used. Customers of IBOD transfer their IT environment to a utility  
management infrastructure (UMI) and pay only for the actual computing services  
received. Like electricity, water and gas, IT is treated as another utility. Thus, by  
eliminating the responsibility of building and maintaining IT operations, providing  
necessary education and training to administrators, and having to comply with  
25 applicable regulations, the customer can focus on their core business while enjoying  
variable pricing, automated processes and the invaluable resilience and  
responsiveness of a shared infrastructure provided by IBM.

[0003] Customers transitioning from a dedicated IT environment to an EBOD  
environment may be concerned about the efficiency of the new paradigm and their  
30 loss of control of actual IT operations. Although current vendors of shared IT  
infrastructure may allow customers to analyze the responsiveness of their IT system,

there are no real-time service testing tools that support the new emerging utility-like middleware such as The IBM EBOD infrastructure.

[0004] Existing products include typical system testing suites, WinRunner published by Mercury Interactive, Inc. of Sunnyvale, California and the forthcoming  
5 Blue Typhoon graphical user interface (GUI) by IBM. Typical system testing suites simply employ computer commands such as “top,” “sar,” “wait,” “vmstat,” “netstat” and “ping” to determine utilization of resources such as CPU, memory, disks and network but do not provide a way to test hypothetical loads.

[0005] WinRunner captures actual IT events and then, based upon the captured  
10 events, emulates users performing the work of maneuvering through a series of GUIs, either client or web-based. The number of users can be adjusted but WinRunner only test hits on application specific functionalities requiring some type of user interface. In other words, WinRunner tests user interfaces rather than simulating the stress on a particular application and the use of specific shared capabilities like processing,  
15 network functionality and data storage.

[0006] Blue Typhoon GUI is a Java swing/applet application that enables customers to modify computing capacity and allocate and/or de-allocate computing resources. Although this approach provides customers with a passive view of resources, Blue Typhoon does not provide a way to retrieve share system/application  
20 data in an active manner with real-life, real-time testing.

Summary of the Invention

[0007] Provided is an apparatus and method for modelling the efficiency and effectiveness of a customer's information technology (IT) system and applications operating in a shared IT environment. The claimed subject matter enables a Universal Management Infrastructure (UMI) provider to demonstrate whether or not an actual customer's request will match or exceed limits of the customer's service level agreement (SLA). Desired IT settings are able to be simulated in the UMI environment by means of a test suite. The test suite enables the customers to generate a production-level load and stress on a virtual computer, thus gaining insight into their particular e-business on-demand (EBOD) environment. This capability enables the customer to identify possible system flaws in advance so that necessary adjustments can be made to maximize the efficiency of their subscribed services. In other words, the claimed subject matter enables the customer to configure and perform real-time testing of their EBOD environment.

[0008] The IBM UMI environment consists of resources such as processing, network and data storage that can handle a large number of users and transactions in the system. A particular client typically only needs a small fraction of this capacity and contracts for a desired level of service. Using helpdesk software such as Tivoli Service Desk (TSD) provided by IBM as an example, the claimed subject matter is able to simulate accesses to detect system response, detect whether or not endpoint machines are optimizing resources such as network/storage capacity and transmit a load representative of that which an typical end user (help desk agent) would perform on a daily basis. In this manner, a customer can analyze and understand the behavior of the virtual computer and, more importantly, the behavior of their company's critical applications within the UMI.

[0009] The present invention enables the customer to compare a hypothetical load against a SLA so that adjustments can be made, if necessary, to manage or optimize the SLA. In addition, prior to the formation of a SLA, a potential customer can be assured that a proposed SLA is optimized and meets their needs.

[0010] This summary is not intended as a comprehensive description of the claimed subject matter but, rather, is intended to provide a brief overview of some of

the functionality associated therewith. Other systems, methods, functionality, features and advantages of the invention will be or will become apparent to one with skill in the art upon examination of the following figures and detailed description.

Brief Description of the Figures

5 [0011] A better understanding of the present invention can be obtained when the following detailed description of the disclosed embodiments is considered in conjunction with the following drawings.

[0012] Figure 1 is a block diagram of an exemplary Universal Management Infrastructure (UMI) architecture incorporating the claimed subject matter.

[0013] Figure 2 is a block diagram an On-Demand Service (ODS) Framework the implements the claimed subject matter.

10 [0014] Figure 3 is a block diagram of an integration hub, introduced in Figure 2, in more detail.

[0015] Figure 4 is a block diagram illustrating the interaction among the integration hub and other various other components of the ODS framework of Figure 2.

15 [0016] Figure 5 is a block diagram of a test center component of the ODS framework of Figure 2.

[0017] Figure 6 is a flow chart of a simulation process that generates a prediction of an impact a specified workflow has on system resources and then compares the prediction to a client's service level agreement.

20 [0018] Figure 7 is a flow chart of an exemplary simulation operation process, which illustrates a portion of the simulation process of Figure 6 in more detail.

Detailed Description of the Figures

[0019] Although described with particular reference to a UMI environment, the claimed subject matter can be implemented in any information technology (IT) system in which load and/or stress testing is desirable. Those with skill in the computing arts will recognize that the disclosed embodiments have relevance to a wide variety of computing environments in addition to those described below. In addition, the methods of the disclosed invention can be implemented in software, hardware, or a combination of software and hardware. The hardware portion can be implemented using specialized logic; the software portion can be stored in a memory and executed by a suitable instruction execution system such as a microprocessor, personal computer (PC) or mainframe.

[0020] Turning now to the figures, Figure 1 is a block diagram of an exemplary Universal Management Infrastructure (UMI) architecture 100 incorporating the claimed subject matter. An enterprise 101 services a number of customers, such as a customer\_1 103, a customer\_2 104, a customer\_3 105, a customer\_4 106 and a customer\_5 107. Enterprise 101 also has relations with a number of suppliers, a supplier\_1 113, a supplier\_2 114 and a supplier\_3 115. For the sake of this example, the particular type of business engaged in by enterprise 101 is not specified because UMI architecture 100, as well as the claimed subject, matter can be applied to practically any type of business that employs an information technology (IT) infrastructure. In fact, UMI architecture 101 can even apply to a hypothetical business that does not have customers and/or suppliers.

[0021] In this example, suppliers 113-5 provide parts and services 121 to enterprise 101 and customers 103-7 purchase products, or offerings, 119. Enterprise 101 includes a business process\_1 123, a business process\_2 124 and a business process\_3 125 to enable enterprise 101 to convert parts and services 121 into offerings 119. Examples of types of business processes include, but are not limited to, a manufacturing supply system, an accounting system, a billing system, a customer management system and a payroll system. The specific number of customers 103-7, suppliers 113-5 and business processes 123-5 are used for the sake

of an example only; the claimed subject matter applies equally well to small, medium and large enterprises with any particular number of such relationships.

[0022] Enterprise 101 incorporates a virtualized infrastructure, or an “On-Demand services (ODS) Framework,” 129, which, in this example, is an e-business on demand (EBOD) environment designed by International Business Machines Corp. (IBM), of Armonk, New York. The IBM EBOD environment is designed for business customers and can deliver accounting, human resource, and customer relationship management applications over the Internet for a usage-based charge, or can be used to provide computing resources, e.g. processors, storage, memory, to a company as needed to support their operation.

[0023] Figure 2 is a block diagram of ODS Framework 129 of Figure 1 in more detail. Included in Figure 2 is an ODS block 167 representing the various on-demand services that may be available in an ODS environment such as the IBM EBOD environment. As mentioned above, examples of ODS services include, but are not limited to, a manufacturing supply system, an accounting system, a billing system, a customer management system and a payroll system. In this example ODS services 167 are coupled to ODS framework 129 via a service programming interface (SPI) 165. In this example, SPI 165 is a set of application programming interfaces (APIs). Those with skill in the computing arts should recognize that there are other ways to implement the connection between ODS block 167 and ODS framework 129 other than via SPI 165, such as but not limited to secure sockets.

[0024] Also included in Figure 2 is a business systems block 169, which represent any or all particular business process 123-5 (Fig. 1) that may be required to provide access to one or more of the various ODS services offered by enterprise 101 (Fig. 1). Business systems 169 is coupled to ODS framework 129 via an order enable block 171, which can represent software, hardware or human operators for communicating information from business systems to ODS framework 129.

[0025] ODS framework 129 includes an integration hub 141 for coordinating the interactions among business system 169, ODS services 167 and ODS framework 129. Integration hub 141 includes a workflow component 143 and an integration middleware component 145. Workflow component 143 manages communication and requests from business systems 169 and integration middleware component 145

communication and requests from ODS block 167. While workflow component 143 communicates primarily with integration middleware component 145, integration middle ware component 145 is responsible for handling communication and requests from both workflow component 143 and ODS services block 167 to and from a UMI base services 147 to ODS component 167.

[0026] UMI base services 147 include a portal 151, which is a communications interface between UMI base services 147, the rest of ODS framework 129 and any entity, such as software from another vendor, that is external to ODS framework 129 and requires a direct communication link to UMI base services 147. Those with skill in the computing arts will realize there are a number of methods of implementing portal 151, including but not limited to, APIs and secure sockets. Additional components of UMI base services 147 include a help desk component 152, a service level agreement (SLA) component 153, a provisioning component 154, a reporting component 155, a Monitoring and Management component 156, a billing component 157, a metering component 158 and a test center component 159.

[0027] Help desk component 152 may be either an automated system such as a typical telephone response system or a fully or partially human staffed system in which help desk component 152 serves to automate communication and data retrieval tasks for employees that work at a corresponding help desk department of enterprise 101 (Fig. 1).

[0028] Service level agreement (SLA) management component 153 monitors and controls the interactions between OSD framework 129 and the client enterprise. Interactions include access to system resources by customers 103-7 (Fig. 1) and/or suppliers 113-5 (Fig. 1). A SLA is typically a contractual agreement between the provider of ODS framework 129 and the enterprise concerning the amount of resources of ODS framework 129 to which the enterprise is entitled and the cost of those resources. In other words, SLA management component 153 determines whether or not enterprise usage and UMI services are meeting, exceeding or otherwise complying with their specific SLA and then takes appropriate actions based upon that information. Data concerning SLAs is stored in a data store 161.

[0029] Provisioning engine 154 provides for the automation of tasks and the distribution of resources related to the resources within ODS framework 129.



Specifically, provisioning engine **154** enables the allocation of resources such as servers, data storage, network resources and firewalls to meet enterprise demand as required by an SLA. In addition, provisioning engine **154** facilitates the distribution of software within the ODS framework **129**.

5   **[0030]**     Reporting component **155** is responsible for the production of reports on any or all of enterprise **101**, business processes **123**, **125** and **127** and ODS framework **129**. Reports may include, but are not limited to, production reports, billing reports, inventory reports, customer reports, performance reports and SLA compliance reports. Pre-defined report templates and generated reports are stored in  
10   data store **161**.

**[0031]**     Monitoring and Management (M&M) component **156** is responsible for the collection of information on and provides the interface for the management of ODS framework **129** and the other UMI base services **147**. Collected information is stored in data store **161** and is made available, either directly or through data store  
15   **161**, to help desk **152**, reporting component **155** and a billing component **157**, explained below.

**[0032]**     Billing component **157** produces invoicing and billing information for the enterprise for its use of ODS framework **129**, based primarily on information from SLA management component **153**, and a metering component **158**, described below.

20   **[0033]**     Metering component **158** keeps track of enterprise use of ODS framework **129**, as well as any necessary internal information relative to the operation of ODS framework **129**. Information collected by metering component **158** is stored in data store **161** and available for use by help desk component **152**, reporting component **155**, M&M component **156** and billing component **157**.

25   **[0034]**     Finally, test center component **159** controls such activities as customer profiling, test data generation, and test storage and scheduling for ODS framework **129**. Test center component **159** is explained in more detail below in conjunction with Figure 5.

**[0035]**     Figure 3 is a block diagram of integration hub **141** of Figure 2 in more  
30   detail, with particular attention paid to communication between particular components of ODS framework **129** during processing of SPI **169** requests. As explained above in conjunction with Figure 2, communication both to and from ODS

block **167** is transmitted via **SPI 165**. A specific request from **ODS 167** via **SPI 165** is transmitted to integration middleware **145**, which then determines the appropriate component to handle the request. Possibilities include a central authentication and authorization (CAA) component **173**, which ensures that the specific request is from  
5 an authorized source, and a common login and audit (CLA) component **175**, which, once a particular source has been authenticated by **CAA 173**, provides a login and records transactions.

[0036] Requests that arrive to integration hub **141** following authentication by **CAA 173** and login by **CLA 175** are then routed to appropriate UMI components  
10 **151-8**. For example, metering component **158** both records a particular user's access by way of **CLA 175** and records the particular users' usage of **ODS framework 129**. As explained above, workflow component **143** regulates integration middleware **145** with respect to the management of transactions within **ODS framework 129**. Workflow component **143** enables automation of operational processes among UMI  
15 components **151-8**. Workflow component **143** also coordinates processes that are partly manual steps and partly automated steps. For manual steps, workflow component **143** performs tasks such as, but not limited to, tracking progress, enforcing time limits and sending alerts based on preset business rules.

[0037] Figure 4 is a block diagram that illustrates the interaction among  
20 integration hub **141**, UMI components **151-8**, **ODS 167**, **CAA 173** and **CLA 175** in more detail, all of which were explained above in conjunction with Figures 2 and 3. **ODS 167** communicates with integration hub **141** by sending requests **181** and receiving responses **183**. For the sake of simplicity, **SPI 165** (Fig. 3) is not shown in Figure 4.

25 [0038] An initial **ODS request 181** from a particular user, such as a customer **103-7** or a supplier **113-5** (Fig. 1), typically needs to be authenticated by **ODS framework 129** (Fig. 2). This is accomplished by integration hub **141**, which extracts an **ODS identification (ID)** from a digital certificate in the initial request **181** and transmits **ODS ID 185** to **CAA 189**. Typically, the **ODS ID**, which can be, but is not limited to,  
30 a password, is established when **ODS 167** is setup for a particular client. **CAA 173** then employs the digital certificate for authentication and authorization. Basically,

CAA 173 performs authentication and authorization based upon an ODS ID/SPI mapping 191 retrieved in response to a request 189 to data store 161.

[0039] If the particular user is authenticated, then CAA 173 transmits an ODS authorization 187 back to integration hub 141. Once integration hub 141 has received this authorization, integration hub 141 is able to send and receive requests and responses 193 from UMI components 151-8 on behalf of the authenticated user. For the sake of simplicity, only billing component 157 illustrates requests and responses 193 broken out into separate paths representing requests 197 and responses 199. Finally, the authentication and authorization activity is logged 195 to CLA 175.

[0040] Figure 5 is a block diagram of test center 159 of Figures 2, 3 and 4 in more detail. Also included in Figure 5, is data store 161, illustrating exemplary data areas relevant to the claimed subject matter. Test center 159 includes a resource profiler component 201, a workload profiler component 203, a simulation engine 205, a simulation data generator 206, a profile comparator 207 and a compliance monitor 209.

[0041] Resource profiler component 201 compiles resource profile data 211, which is stored in data store 161. Resource profile data 211 represents a particular client's resources allotments in the ODS framework 129 as well as any other available resources. Examples of allocated resources might include, but are not limited to, processing cycles, number of required servers, network bandwidth and data storage requirements. Each of the allocated resources have parameters associated with them, such as a base resource allotment, a maximum resource allotment, a resource cost and rules for dynamically reallocating the resources based upon the client's workload demand. An example of an available resource is the types of available processors. For example, a user might have contracted to use an Intel architecture but have a need to know how their applications function in a more robust server environment.

[0042] Workload profiler component 203 generates workload profile data 213, also stored in data store 161. Workload profile data 213 represents a particular client's typical workload with respect to the client's allocated resources as described in the customer's resource profile data 211, or workload with respect to allocations that use resources. For example, business process\_1 123 of enterprise 101 (Fig. 1)

may typically generate one hundred (100) calls per day to billing component 157 of ODS framework 129 (Fig. 2), with each request 197 averaging one kilobyte and each response 199 averaging two kilobytes. In addition, one transactions may require an average of one hundred (100) processing cycles to generate a response 199 from an average request 197 and generate a one megabyte chunk of data storage 161 (Figs. 2 and 4). This information is stored in workload profile data 213 as a typical workload for business process\_1 123. Each business process 123-5 has data in workload profile data 213 corresponding to its typical transactions and each transaction's utilization of such parameters as processing cycles, bandwidth and data storage.

10 [0043] In addition to workload profile data 213 based upon a particular customer's actual usage, workflow profile data 213 includes data corresponding to a hypothetical client, perhaps based upon a collective or predicted average. In this manner, a potential customer can generate and examine scenarios corresponding to a potential workload, thus either assuring themselves that ODS framework 129 can  
15 handle the business or predicting a particular level of SLA that might be appropriate for the potential client. Workload profile data 213 can also be stored in configuration files so that various hypothetical scenarios are saved for future reference and can be modified if necessary.

[0044] Simulation engine 205 employs workload profile data 213 to generate  
20 hypothetical workloads based upon projected changes in a client's workload entered by the client or administrator via the manipulation of parameters in a GUI (not shown). For example, based upon a figure for an average number of transactions to billing component 157, simulation engine 205 can estimate resources of ODS framework 129 that would be required to increase an average workflow of one  
25 hundred transactions per day to two hundred transactions per day. Simulations can be generated with varying degrees of granularity. For example, a client may know that business process\_1 123 is going to experience a surge in demand on a certain date. In this case, workflow profile data 213 includes information indicating that business process\_1 123 typically executes one hundred (100) transactions against billing  
30 component 157, fifty (50) transactions on reporting component 155, two hundred transactions on metering component 158, and so on. Simulation engine 205 then generates hypothetical workloads corresponding to each particular, applicable

component **151-8**. Various combinations of parameters can be saved by a client so that a particular simulation represented by a specific set of parameters can be rerun or tweaked simply by modifying the stored set of parameters. This feature prevents the unnecessary reentry of parameters.

5   **[0045]**     In addition to a creating an estimation of ODS framework **129** response to a particular hypothetical demand, simulation engine **205** can actually use simulated data to make a physical test on ODS framework **129**. For example, using the example above of one transactions that requires one hundred (100) processing cycles and one megabyte chunk of data storage **161**, simulation data generator **206** generates  
10   appropriately sized dummy programs, network data packets and blocks of simulated data so that simulation engine **205** is a able to actually consume processing cycles, transmit network traffic and allocates blocks of data storage **161** in amounts that equal the hypothetical load. In this manner, the hypothetical workload is simulated using the actual available resources and infrastructure. An exemplary data acquisition  
15   process **240** is described below in conjunction with Figure 7.

**[0046]**     Simulation engine **205** can create hypothetical schedules for predicted or hypothetical workloads so that the effect of timing issues can be factored into the simulation. For example, the client may know that web traffic and corresponding product orders from customers typically peak between 5:00 pm and 6:00 pm and that  
20   report processing can be delayed until after the 5:00 rush. In this case, different types of processing can be segregated from each other and the effect of various segregation schemes can be evaluated.

**[0047]**     In an alternative embodiment, simulation engine **205** can make predictions on hypothetical scenarios of a first client based upon a match between the  
25   hypothetical scenario and actual resource profile data **211** and workload profile data **213** corresponding to a second client whose actual data closely matches the hypothetical scenario.

**[0048]**     Profile comparator **207** compares the hypothetical workloads produced by simulation engine **205** then to resource profile data **211**. In this manner, test center  
30   **159** determines whether or not enough resources exist and whether or not the existing resources are available to service the hypothetical workload. It should be noted that this determination is accomplished without impact on the actual resources of ODS

framework 129 other than perhaps hits on data store 161 and the processing cycles required to perform the simulation itself.

[0049] Compliance monitor 209 takes the results of both simulation engine 205 and profile comparator 209 and determines whether or not projected or hypothetical  
5 workloads either will attempt to utilize unavailable resources or violate the terms of a client's SLA, which is stored in a SLA data portion 215 of data store 161. The ability to create multiple hypothetical workloads and evaluates them with simulation engine 205 enables the client to determine an appropriate SLA for projected business and whether a current SLA is cost efficient.

10 [0050] Once processing of simulation engine 205, profile comparator 207 and compliance monitor 209 is complete, then ODS framework 129 signals the client with the results.

[0051] Figure 6 is a flow chart of a Simulation and Compare process 220 that generates a prediction of the impact a specified workflow has on system resources  
15 and then compares the prediction to a client's service level agreement. Process 220 starts in a "Begin Simulation" step 221 and processing proceeds immediately to a "Retrieve Resource Profile" step 223, in which process retrieves resource profile data 211 (Fig. 5) for a particular client from data store 161 (Figs. 2, 4 and 5). As explained above in conjunction with Figure 5, resource profile data 211 represents a  
20 particular client's resources allotments and associated parameters in the ODS framework 129 (Fig. 2).

[0052] Control then proceeds to a "Retrieve Workload Profile" step 225 in which process 220 retrieves workload profile data 213 corresponding to the client's whose resource profile data 211 was retrieved in step 221. As explained above in  
25 conjunction with Figure 5, workload profile data 213 represents the particular client's typical workload with respect to the client's allocated resources as described in the customer's resource profile data 211. In addition, various parameters may have corresponding importance and/or prioritization attributes that affect a particular simulation.

30 [0053] Once a particular client's resource profile data 211 and workload profile data 213 have been retrieved, control proceeds to a "Modify Workload Profile" step 227 in which the client can modify parameters in the retrieved workflow profile data

213 in order to express hypothetical scenarios, or projected workflow, such as sudden spikes in demand for website access. In other words, by modifying workflow profile data 213, the client can test anticipated changes in the client's actual workflow. Changes to workflow profile data 213 are accomplished via a GUI (not shown), the creation and operation of which should be understood by those with skill in the programming arts. Further, alternative scenarios, and their corresponding workload profile data 213, can be stored in configuration files, loaded into process 220 during step 225 and modified, if necessary, in step 227.

[0054] Once the client has modified the desired parameters in step 227, the control proceeds to a "Simulate Operation" step 229 in which process 220 simulates the projected workflow as performed on the resources defined in the client's resource profile data 211. Simulation operation step 229 is explained in more detail below in conjunction with Figure 7. Next, control proceeds to a "Compare Operation to SLA" step 231 in which, first, SLA data 215 (Fig. 5) is retrieved from data store 161 and, then, the results of the simulation operation performed in step 229 is compared to the clients SLA data 215.

[0055] Control then proceeds to a "SLA Sufficient?" step 233 in which process 220 determines whether or not the projected workflow as calculated in step 229 exceeds the client's limits with respect to resources, as defined in the client's SLA. If the projected workflow does not exceed to client's SLA, then the client is notified of that fact and control proceeds to an "End Simulation" step 237 in which processing is complete. If in step 335 the projected workflow exceeds the client's limits as defined by the SLA, then control proceeds to a "Generate New SLA" step 235 in which SLA data 215 is modified so that SLA data 215 conforms with the results of the simulation. In addition, process 220 may interact with billing component 157 (Figs. 2 and 4) in order to calculate a cost associated with the needed level of service predicted by the simulation.

[0056] The modified SLA data 215 is then presented to the client so that the client can determine whether or not to modify their SLA so that the ODS framework 129 can handle the expected workload within the agreement. Multiple possible conforming SLAs, along with their respective cost, may be proposed to the client based upon a particular simulation. For example, the simulation may raise the

possibility that a client's business may be better served by switching from an Intel architecture agreement to a server based agreement. In the alternative, process 220, instead of predicting a needed level SLA, predicts a maximum work load a particular SLA can tolerate before the client is in violation of the agreement.

5 [0057] Figure 7 is a flowchart of exemplary simulation operation process 240, corresponding to simulation operation step 229 of Figure 6. Process 240 starts in a "Begin Simulation Operation" step 241 and control proceeds immediately to a "Determine Source" step 243. In step 243, process 240 scans historical simulation data in order to determine whether or not, with respect to the hypothetical scenario,  
10 there is sufficient data, either from the current client or other similar clients, to estimate load and the load's impact on affected resources of ODS framework 129. The load and the load's impact are two different aspects of historical data and the sufficiency of both must be addressed in the next step.

[0058] If there is sufficient historical data, then control proceeds to a "Retrieve Data" step 247 in which the historical data is retrieved from data store 161. If not,  
15 then control proceeds to a "Generate Data" step 249 in which process 240 creates relevant test data for the next step of the process. Relevant test data may be actually simulated data such as dummy client files and transactions or simply blocks of random data of an appropriate size. Of course, if there is some historical data but not  
20 enough to provide a meaningful simulation, control would proceed from step 245 to step 249 and only the data that is needed would be generated.

[0059] Once data has been acquired, either retrieved data in step 247, generated data in step 249 or some combination of the two, then control proceeds to an "Actual Resources?" step 251 in which process 240 determines whether or not to calculate  
25 the simulated data's load on ODS framework 129 or actually generate the load on the resources of framework 129. This decision can be based on either an administrator's or user's preference. For example, an administrator may determine that simulations are permitted to utilize actual resources only during off-hours and that during other times only calculated simulations can be run.

30 [0060] If, in step 251, process 240 determines that actual resources are to be utilized in the simulation, then control proceeds to an "Execute Simulation" step 255 in which the retrieved and/or generated data is actually used to transmit packets,



generate processing cycles, occupy a portion of data store 161, and so on. During step 255 relevant information is collected to determine the load's impact on the resources of ODS framework 129. If, in step 251, process 240 determines that actual resources are not to be used, then control proceeds to a "Calculate Simulation" step in  
5 which the relevant information on the load's impact of ODS framework 129 is extrapolated based upon the retrieved and/or generated data. Finally, from both steps 253 and 255, control proceeds to an "End Simulation Operation" step 257 in which process 240 is complete.

[0061] While the invention has been shown and described with reference to  
10 particular embodiments thereof, it will be understood by those skilled in the art that the foregoing and other changes in form and detail may be made therein without departing from the spirit and scope of the invention, including but not limited to additional, less or modified elements and/or additional, less or modified steps performed in the same or a different order.